

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Matevž Mrak

**Upravljanje drsnih vrat na daljavo z vgrajenim sistemom
in mobilno aplikacijo**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

Ljubljana 2016

Rezultati diplomskega dela so intelektualna lastnina avtorja. Za objavljane ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

V današnjem času uspevamo s pomočjo sodobnih tehnologij poenostaviti veliko različnih procesov. Kontrola dostopa do oddaljenih objektov je tipičen primer takega procesa. Izdelajte prototip takšnega sistema za upravljanje drsnih vrat na daljavo s pomočjo SMS sporočil. Izberite gradnike vgrajenega sistema in ga povežite z enim od trenutno dostopnih motorjev za upravljanje z drsnimi vrati. Pri tem poskrbite za varnost uporabe takega sistema z različnih vidikov. Naredite tudi mobilno aplikacijo, ki bo olajšala upravljanje sistema. Uvedite tudi dve vrsti uporabnikov z različnimi dostopnimi pravicami. Sistem naj omogoča tudi dinamično dodajanje uporabnikov in v čim večji meri tudi nadaljnje nadgradnje. Pri gradnji uporabite cenovno dostopne tehnologije in poskušajte doseči čim ugodnejšo ceno prototipnega sistema.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Matevž Mrak sem avtor diplomskega dela z naslovom:

Upravljanje drsnih vrat na daljavo z vgrajenim sistemom in mobilno aplikacijo

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom viš. pred. dr. Roberta Rozmana,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,

V Ljubljani, dne 22. februarja 2016

Podpis avtorja:

Rad bi se zahvalil svojim staršem, dekletu in prijateljem, ki so me podpirali tekom študija ter med izdelavo diplomske naloge. Posebej bi se rad zahvalil mentorju viš. pred. dr. Robertu Rozmanu za vso pomoč in uporabne nasvete.

Kazalo

Povzetek

Abstract

Poglavje 1	Uvod	1
Poglavje 2	Aparaturni gradniki sistema.....	3
2.1	Motor za drsna vrata Came BX-78.....	4
2.1.1	Kontrolna plošča.....	5
2.1.2	Priključitev in konfiguracija	7
2.2	Vgrajeni sistem za upravljanje vrat	10
2.2.1	Mikrokrmilniški sistem Arduino UNO.....	11
2.2.2	Komunikacijski modul GSM/GPRS za Arduino UNO	12
2.2.3	Modul z dvokanalnim relejem.....	12
2.2.4	Tipalo položaja vrat – »Reed« stikalo	13
2.2.5	Modul z bralnikom SD-kartic.....	14
Poglavje 3	Pregled programskih tehnologij in orodij	15
3.1	Razvojno okolje Arduino IDE – C++	15
3.2	Razvojno okolje Android Studio – Java, XML, HTML in Sugar ORM	16
Poglavje 4	Funkcionalnost sistema	18
4.1	Uporabnik aplikacije.....	18
4.2	Skrbnik sistema.....	20
Poglavje 5	Izdelava prototipa sistema	21
Poglavje 6	Programska oprema vgrajenega sistema.....	24
6.1	Upravljanje vrat	24
6.2	Komunikacija z mobilnim omrežjem	25
6.3	Branje in izvajanje ukazov uporabnika	27

6.4 Branje in izvajanje ukazov skrbnika	28
Poglavje 7 Mobilna aplikacija za upravljanje drsnih vrat.....	30
7.1 Uporabniški vmesnik mobilne aplikacije.....	30
7.2 Dodajanje, urejanje in brisanje naprav.....	30
7.3 Upravljanje naprav.....	31
Poglavje 8 Sklepne ugotovitve.....	35
Poglavje 9 Literatura	37

Seznam uporabljenih kratic

kratica	angleško	slovensko
SMS	Short Message Service	Kratko besedilno sporočilo
GSM	Global System for Mobile Communications	Standard mobilnih komunikacij
NO	Normally Open	V mirovnem stanju razklenjen kontakt
NC	Normally Closed	V mirovnem stanju sklenjen kontakt
GPRS	General Packet Radio Service	Splošna paketna radijska storitev

Povzetek

Naslov: Upravljanje drsnih vrat na daljavo z vgrajenim sistemom in mobilno aplikacijo.

Danes se avtomatizacija procesov vse pogostejše uporablja praktično na vseh področjih. Avtomatizacija pristopov k objektom je prav gotovo eden od takih primerov, ki ga je mogoče realizirati na več različnih načinov. V sklopu diplomske naloge smo izdelali prototipni vgrajeni sistem za krmiljenje motorja drsnih vrat proizvajalca Came, ki se lahko upravlja na daljavo s pomočjo SMS sporočil. Vgrajen sistem temelji na mikrokrmilniškemu sistemu Arduino UNO in komunikacijskem modulu GSM. Upravljanje s sporočili SMS smo dodatno poenostavili s pomočjo mobilne aplikacije za pametne telefone in tablice, na katerih teče operacijski sistem Android. Aplikacija omogoča več funkcij upravljanja drsnih vrat na daljavo, kot so odpiranje in zapiranje vrat, preverjanje njihovega stanja in povezav s samo napravo.

Ključne besede: Arduino, vgrajeni sistem, mikrokrmilnik, Android, drsna vrata, motor.

Abstract

Title: Remote Sliding Door Control by Embedded System and Mobile Application.

Nowadays, automation of processes is increasingly used in virtually all areas. Automatic control of access to remote buildings is typical such case that can be implemented in many different ways. Within the scope of this diploma thesis, we have developed a prototype embedded system used for controlling the remote sliding door motor by means of SMS messages. The embedded system is based on the microcontroller Arduino with GSM communication module. SMS messages management was further simplified by the application for Android smart phones. The application enables various remote control functions, such as closing and opening the door, checking door's condition and connections with the device itself.

Keywords: Arduino, embedded system, microcontroller, Android, sliding door, motor.

Poglavje 1 Uvod

V podjetju Palisada d.o.o., Jesenice se med drugim ukvarjajo tudi s prodajo in montažo motoriziranih vrat. Podjetja, ki dobavljajo motorje, trenutno omogočajo odpiranje vrat z oddajnikom, brezstično kartico, induktivno zanko, s tipko in SET-ključavnico [1]. Podjetje Palisada d.o.o., Jesenice ima skladiščne prostore na lokaciji, ki je od sedeža podjetja oddaljena osem kilometrov. Ob vsakem obisku neavtoriziranih oseb v območje skladišča se morajo na lokacijo pripeljati in osebno poskrbeti za odpiranje vrat, kar jim vsakodnevno vzame veliko dragocenega časa. Glede na to, da dandanes večina obiskovalcev uporablja pametne telefone, smo se odločili, da razvijemo sistem, ki bi omogočal odpiranje vrat na daljavo. Po preučitvi vseh možnosti smo uporabili mikrokrmilniški sistem Arduino skupaj z modulom GSM, ki med drugim omogoča tudi pošiljanje in prejemanje SMS-sporočil. Komunikacijo s SMS-sporočili smo izbrali, ker na veliko objektih ni moč uporabljati brezžične internetne povezave ali signala GPRS.

Podjetje je poskrbelo za nakup motorja, nato smo izdelali prototip in z njim povezali vse elemente motorja. To nam je omogočilo testiranje v notranjih prostorih. Nemoteno delovanje motorja je bil pogoj, da smo nanj lahko priključili vgrajeni sistem. Najprej smo namesto stikal, ki se uporabljata za odpiranje ter zapiranje vrat z motorjem povezali dva releja, ki sta v našem primeru nadomestili stikali. Če se ne nahajamo v bližini drsnih vrata je pred upravljanjem potrebno vedeti ali so vrata odprta ali zaprta. Ker iz kontrolne plošče motorja ne moremo pridobiti podatka o stanju vrat, smo na Arduino vezali tipalo položaja vrat (»reed stikalo«), ki je nameščen na zunanji strani motorja. Tako bi v primeru, da je na konec vrat pritrjen magnet, stikalo sklenil. Ker je najbolj pogosta težava, da vrat ni mogoče odpreti ali zapreti prekritje fotocelic, smo sistem sprogramirali tako, da nas v primeru težav o tem obvesti.

Ko so nam delovale vse osnovne funkcije upravljanja vrat smo se lotili priključitve modema GSM. Po uspešnem testu prejemanja SMS-sporočil smo določili kakšne ukaze bo naš vgrajeni sistem sprejemal in jih posredoval kontrolni plošči motorja. Nato smo sprogramirali logiko za pošiljanje povratnih informacij uporabniku o stanju vrat, prekritosti fotocelic ter možnostih upravljanja. Ker bi v tem primeru lahko pošiljali ukaze vsi, ki imajo možnost pošiljanja SMS-sporočil, smo se odločili, da bomo na spominsko SD-kartico shranili telefonske številke, ki imajo dovoljenje pošiljanja ukazov našemu sistemu. Številke smo shranili v tekstovno datoteko in prva med njimi je mobilna številka skrbnika sistema, ki ima edini možnost dodajanja ter brisanja uporabniških telefonskih števil. Prav tako lahko svojo telefonsko

številko nadomesti s telefonsko številko drugega skrbnika. S tem smo zaključili programiranje mikrokrmilniškega sistema.

Ker je pošiljanje sporočil napravi precej nepraktično smo se odločili, da bomo razvili mobilno aplikacijo za telefon, ki uporabljajo operacijski sistem Android. Najprej smo sprogramirali dodajanje ter brisanje novih naprav, kar nam omogoča uporabo več različnih naprav. Ob kliku na posamezno napravo se nam pokaže meni z možnostjo izbrisa, urejanja ali upravljanja naprave. Uporabnik ima možnost upravljanja z napravo in sicer lahko s pritiski na gumbe s poslanimi SMS-sporočili v obliki ukazov odpira ter zapira vrata, preveri stanje vrat ter fotocelic. Ob pritisku na gumb dobi potrdilo, da je bilo SMS-sporočilo poslano napravi in v manj kot desetih sekundah dobi odgovor na poizvedbo.

Drugo poglavje diplomske naloge predstavlja gradnike vgrajenega sistema ter njihovo vezavo na motor. V tem poglavju je moč najti tudi podatke o mikrokrmilniškemu sistemu Arduino, ter dodatnih modulih, ki smo jih uporabili pri realizaciji projekta. Tretje poglavje je namenjeno predstavitvi programske opreme ter jezikov, s katerimi smo sprogramirali rešitve. V četrtem poglavju je opisana funkcionalnost sistema. Peto poglavje je podroben opis prototipa sistema in njegove izdelave, v šestem pa predstavljamo programiranje vgrajenega (mikrokrmilniškega) sistema. Naslednje poglavje opisuje izdelavo uporabniškega vmesnika ter programiranje funkcij mobilne aplikacije za operacijski sistem Android, ki omogoča upravljanje z več napravami.

V zaključnem poglavju so opisane težave in možne nadaljnje izboljšave vgrajenega sistema in mobilne aplikacije za upravljanje z drsnimi vrati.

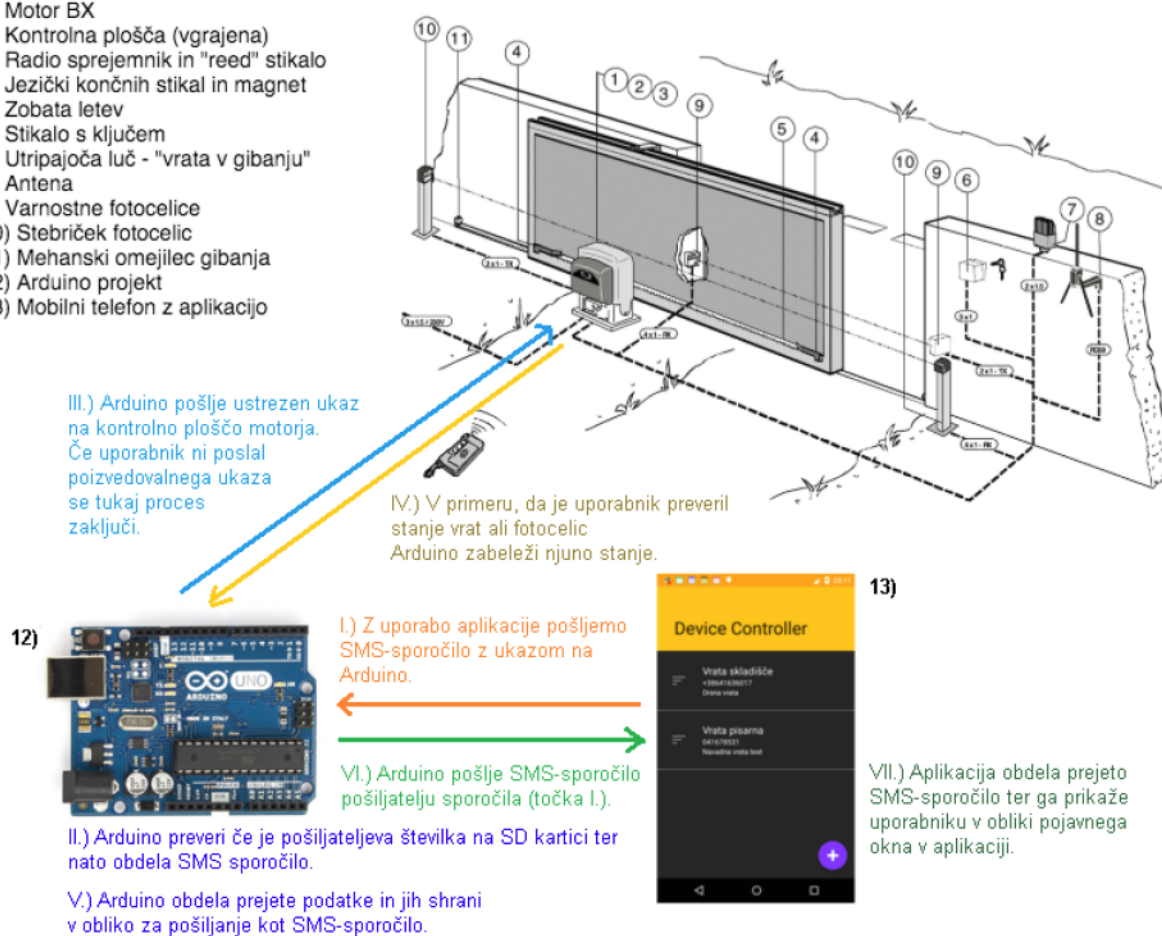
Poglavje 2 Aparaturni gradniki sistema

Za izdelavo prototipa smo uporabili motor BX-78 proizvajalca Came, ki je najpogosteje uporabljen motor pri izvedbi drsnih vrat v podjetju Palisada d.o.o.. Motor krmilimo z vgrajenim sistemom, ki ga namestimo v notranji del motorja pod kontrolno ploščo. Za potrebe komunikacije med telefonom in motorjem smo uporabili modul GSM za Arduino UNO. Uporabili smo funkcijo za pošiljanje ter prejemanje SMS-sporočil, modul pa ponuja veliko več ostalih funkcij, ki v našem primeru niso prišle v poštev. Ker na začetku projekta nismo vedeli kakšen obseg funkcij bi potrebovali, smo se odločili za modul z več funkcijami, ki lahko služi tudi nadaljnjim širitvam našega sistema.

Motor upravljamo z dvema NC-relejema, ki ju uporabimo kot mirovno sklenjen kontakt, in nadomeščata tipki s katerima uporabniki vrat lahko upravljajo z motorjem. Mirovno razklenjen kontakt je potreben zato, ker ob pravilni vezavi relejev na motor, ter poslanem signalu iz digitalnega priključka Arduina na rele, se ta kontakt sklene in kontrolni plošči pošlje signal za odpiranje ali zapiranje vrat. Prvi rele bo služil popolnemu odpiranju in zapiranju vrat, drugi rele pa delnemu odpiranju vrat za pešce.

Pomanjkljivost, da na kontrolni plošči motorja ni mogoče ugotoviti ali so vrata odprta ali zaprta, smo odpravili tako, da smo nanj namestili »reed« stikalo, ki se ob zaprtju prekrije z magnetom, nameščenim na vratih. Za odpiranje in zapiranje vrat lahko uporabimo tudi stikalo, ki ga upravljamo s ključem.

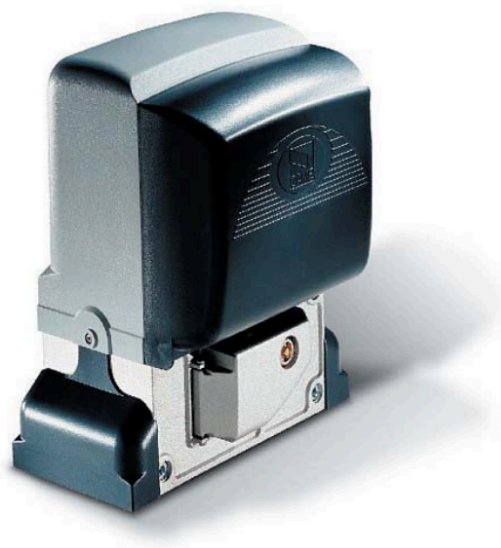
- 1) Motor BX
- 2) Kontrolna plošča (vgrajena)
- 3) Radio sprejemnik in "reed" stikalo
- 4) Jezički končnih stikal in magnet
- 5) Zobata letev
- 6) Stikalo s ključem
- 7) Utripajoča luč - "vrata v gibanju"
- 8) Antena
- 9) Varnostne fotocelice
- 10) Stebriček fotocelic
- 11) Mehanski omejilec gibanja
- 12) Arduino projekt
- 13) Mobilni telefon z aplikacijo



Slika 2.1: Prikaz zgradbe in delovanja sistema [2,7].

2.1 Motor za drsna vrata Came BX-78

Motor za drsna vrata Came BX-78 omogoča odpiranje in zapiranje vrat. Ponuja tudi možnost priključitve fotocelic, ki ob prekritju omogočajo popolno zaustavitev vrat ali ponovno odpiranje. Na motor lahko priključimo signalno luč, ki utripa, ko se vrata zapirajo ali odpirajo. Na stranicah vrat sta nameščeni mehanski zapori, ki v primeru, da pride do okvare končnih stikal, poskrbita, da se vrata na zeleni točki nameščene zapore ustavijo. Sestavni del motorja je tudi antena, ki služi za prejemanje signalov daljinskega oddajnika za upravljanje z vrati.



Slika 2.2: Motor Came BX – 78 [2].

Ime	Enota
<i>Napetost</i>	230 V (AC)
<i>Moč</i>	300 W
<i>Tok</i>	2,4 A
<i>Največji navor</i>	32 Nm
<i>Potisna sila</i>	800 N
<i>Redukcijsko razmerje</i>	1 / 33
<i>Hitrost</i>	10 m/min
<i>Intenzivnost delovanja</i>	30 %
<i>Teža</i>	15 kg
<i>Največja nosilnost</i>	800 kg
<i>Kondenzator</i>	20 μ F
<i>Maksimalna dovoljena temperatura</i>	55 °C
<i>Minimalna dovoljena temperatura</i>	-20 °C

Tabela 2.1: Tehnične karakteristike motorja [2].

2.1.1 Kontrolna plošča

Kontrolna plošča se na kontaktih L in N (prikazana na sliki 2.5) napaja z napetostjo 230V. Zaščitena je s 5A varovalkami. Vsi ostali kontrolni sistemi so napajani z napetostjo 24V in

zaščiteni z 1A varovalko. Kontrolno ploščo konfiguriramo s pomočjo DIP-stikal. Prvo 4-polno DIP-stikalo se uporablja za izbiro funkcij, drugo 10-polno DIP-stikalo pa za njihovo nastavitve. V tabelah 2.4 ter 2.5 so opisi vseh funkcij, ki jih lahko nastavimo s pomočjo DIP-stikal.

Številka stikala	Pomen vklopljenega stikala	Pomen izklopljenega stikala
1	Avtomatsko zapiranje vključeno	Avtomatsko zapiranje izključeno
2	Odpiranje, ustavljanje in zapiranje s tipko in oddajnikom	Odpiranje, zapiranje s tipko in oddajnikom
3	Samo odpiranje z oddajnikom	Samo odpiranje mogoče z vsemi ostalimi napravami
4	Vrata se odpirajo samo, ko držimo gumb na oddajniku ali tipki	Funkcija izključena
5	Vključitev utripanje lučke, ko se vrata zapirajo ali odpirajo	Funkcija izključena
6	Ko fotocelice zaznajo predmet se vrata ustavijo	Funkcija izključena
7	Funkcija izključena	Ko fotocelice zaznajo predmet se vrata med zapiranjem, odprejo nazaj do konca
8	Funkcija izključena	Ko fotocelice zaznajo predmet se vrata ustavijo, ko predmeta ne zaznajo več se do konca zaprejo
9	Funkcija izključena	Ko fotocelice zaznajo predmet se vrata ustavijo do posredovanja uporabnika
10	Funkcija izključena	Hitro ustavljanje motorja v končnih pozicijah vključeno

Tabela 2.2: Nastavitve in pomen 10 – polnega DIP stikala [2].

Številka stikala	Pomen vklopljenega stikala	Pomen izklopljenega stikala
1	Samo zapiranje vrat z napravami povezanimi na priključka 2 ter 7 (prikazana na sliki 2.4)	Odpiranje in zapiranje vrat z napravami povezanimi na priključka 2 ter 7 (prikazana na sliki 2.4)
2	Samo odpiranje vrat z napravami povezanimi na priključka 2 ter 3P (prikazana na sliki 2.4)	Delno odpiranje vrat z napravami povezanimi na priključka 2 ter 3P (prikazana na sliki 2.4)
3	Optični čitalec ni aktiviran	Optični čitalec aktiviran
4	Ne uporabi tega DIP stikala	DIP stikalo v uporabi

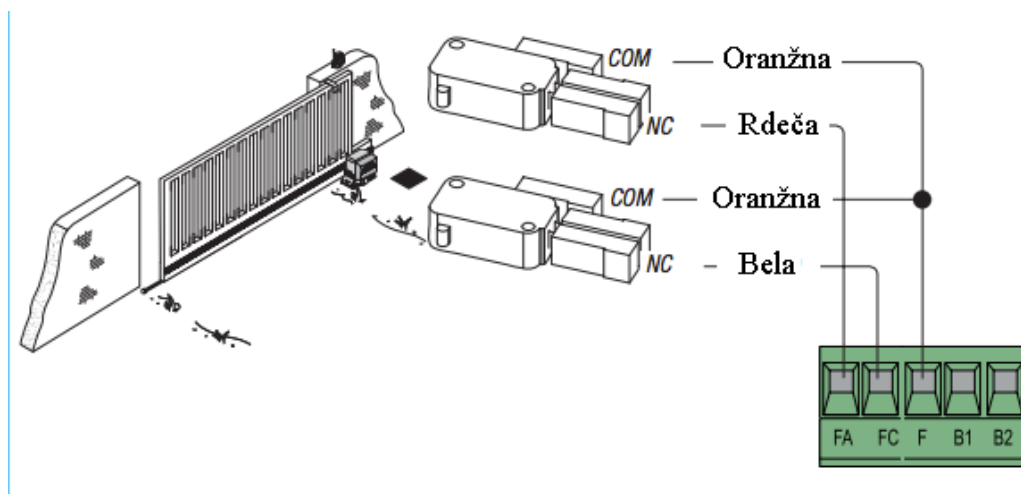
Tabela 2.3: Nastavitve in pomen 4 – polnega DIP stikala [2].

Fotocelice lahko na motor vežemo na več načinov, ki v primeru prekritja pošljejo signal kontrolni plošči, ki lahko izvede naslednje ukaze:

- Ponovno odpiranje vrat, ko so v fazi zapiranja
- Delna ustavitev: vrata se ustavijo, aktivira se avtomatsko zapiranje, če je vključeno
- Popolna ustavitev: vrata se ustavijo, avtomatsko zapiranje se izključi. Vrata ostanejo v tem položaju do ustreznega ukaza s strani uporabnika.

2.1.2 Priključitev in konfiguracija

Na kontakte FA, FC in F (prikazani na sliki 2.3) zvežemo končni stikali, ki služita za zaustavitev vrat, ko le-ta pridejo do končnega položaja (slika 2.3). Končni stikali je potrebno povezati tako, da tvorita mirovno sklenjen kontakt. To pomeni, da se bo ob pritisku vrat na končno stikalo kontakt razklenil, kar bo povzročilo zaustavitev motorja.

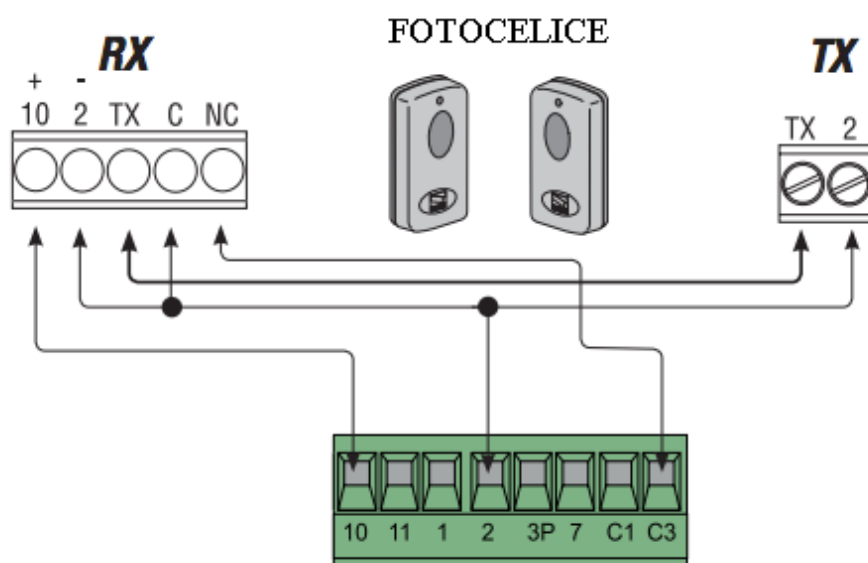


Slika 2.3: Priklop končnih stikal [2].

Luč priklopimo na kontakta W in E1 (prikazana na sliki 2.5). Če želimo, da luč utripa ob odpiranju in zapiranju je potrebno vključiti stikali 1 in 6 na 10-polnem DIP-stikalu (tabela 2.2).

Na kontakte 2, 3P in 7 (prikazani na sliki 2.4), ki so namenjeni priključitvi gumbov za odpiranje in zapiranje vrat s tipkami, priključimo dva releja z mirovno razklenjenim kontaktom (NO).

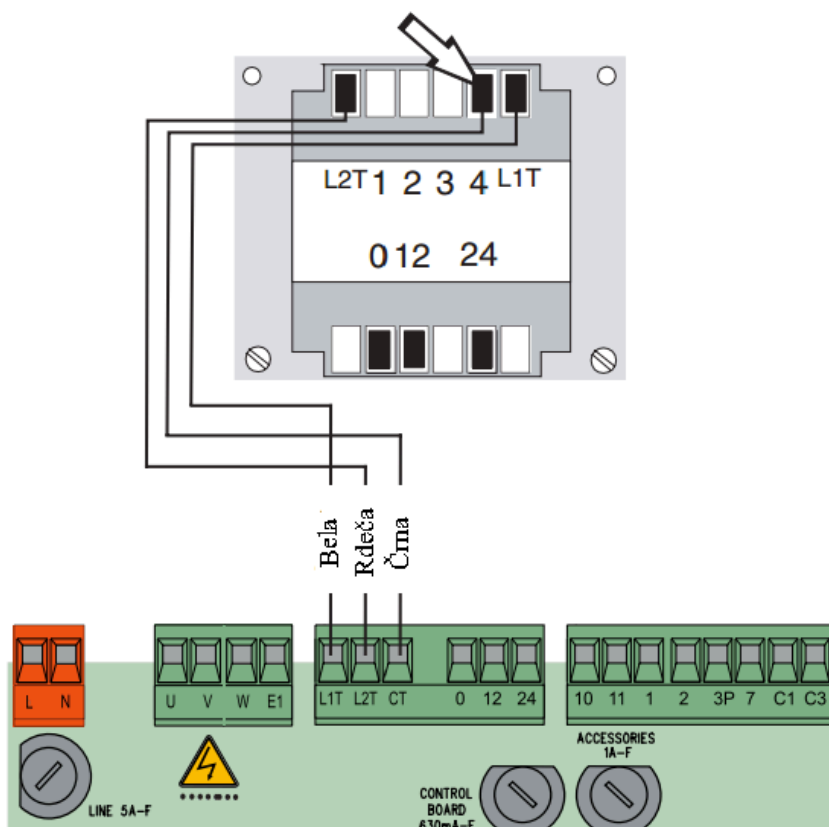
Fotocelice smo priključili tako, da se vrata ustavijo, če celice zaznajo premikanje. Če želimo, da se ob prekritju fotocelic izvede druga funkcija je potrebna drugačna nastavitvev DIP-stikal in priključitev fotocelic. Naše nastavitvev DIP-stikal smo predstavili v tabelah 2.4 in 2.5, njihove funkcije glede na položaj pa so opisane v tabelah 2.2 in 2.3.



Slika 2.4: Priklop fotocelic [2].

Kontrolna plošča omogoča nastavitve dolžine odpiranja vrat v primeru delnega odpiranja. Pri avtomatskem zapiranju lahko nastavimo tudi, po koliko minutah naj se vrata po odprtju avtomatsko zaprejo. Na voljo sta še dva gumba (CH1 in CH2), ki služita za pomnenje oddajnikov.

Obstaja tudi možnost nastavitve navora motorja. Navor lahko nastavljamo od 1 (najmanjši) do 4 (največji), in sicer s priključitvijo črnega kabla na zelen priključek (slika 2.5).



Slika 2.5: Nastavitev navora motorja [2].

Zaporedna številka stikala	Pozicija (ON/OFF)
1	ON
2	ON
3	OFF
4	OFF
5	OFF
6	OFF
7	ON
8	ON
9	ON
10	ON

Tabela 2.4: Nastavitev 10-polnega DIP-stikala za naše potrebe

Zaporedna številka stikala	Pozicija (ON/OFF)
1	OFF
2	ON
3	OFF
4	OFF

Tabela 2.5: Nastavitev 4-polnega DIP-stikala za naše potrebe

2.2 Vgrajeni sistem za upravljanje vrat

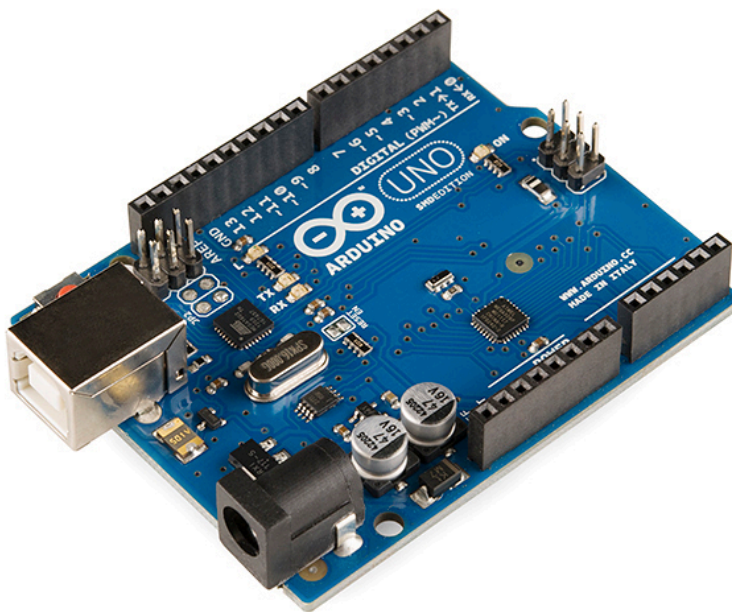
Naš vgrajeni sistem je sestavljen iz več elementov. Sistem bomo vgradili v notranjost motorja. Sestavljen je iz mikrokrmilniškega sistema Arduino UNO, nanj pa so zvezani komunikacijski modul GSM/GPRS, modul z dvo-kanalnim relejem, tipalo položaja vrat ter modul z bralcem SD-kartic.

2.2.1 Mikrokrmilniški sistem Arduino UNO

Mikrokrmilniški sistem Arduino UNO je osnovan na mikrokrmilniku ATmega328P. Ima 14 vhodno/izhodnih priključkov (od tega šest PWM-izhodov) ter 6 analognih vhodnih priključkov. Za napajanje lahko uporabimo USB-povezavo, napajalni adapter ali baterijo. Ima 32kB flash, 2kB SRAM ter 1kB EEPROM spomina. Priporočena vhodna napetost je 12V. Ukaze izvaja s frekvenco urinega signala 16MHz.

Za napajanje mikrokrmilniškega sistema smo uporabili kabel USB in ga priključili na računalnik, ki preko njega tudi komunicira. Arduino je najugodnejše možno naročiti preko spleta, kupiti pa ga je mogoče tudi v trgovinah z elektroniko. Najprej smo želeli uporabiti cenejšo kopijo Arduina, ampak se izkazalo, da ni kompatibilen z modulom GSM, zato smo si priskrbeli originalno različico.

Arduino lahko uporabimo za ogromno različnih projektov. Prav tako pa nam obsežna dokumentacija na spletu omogoča odgovor skoraj na vsako vprašanje. Ker pred tem nismo imeli izkušenj s takšnimi projekti smo se ravno zaradi tega odločili za Arduino.



Slika 2.6: Mikrokrmilniški sistem Arduino Uno [7].

2.2.2 Komunikacijski modul GSM/GPRS za Arduino UNO

Za povezavo z mobilnim omrežjem je potreben modul GSM/GPRS, ki se uporablja skupaj z mikrokrmilniškim sistemom Arduino UNO.

Za potrebe našega projekta smo se odločili za modul GSM/GPRS SIM900 proizvajalca IComSat. Modul upravljamo z AT-ukazi in je kompatibilen z Arduino Uno [4].

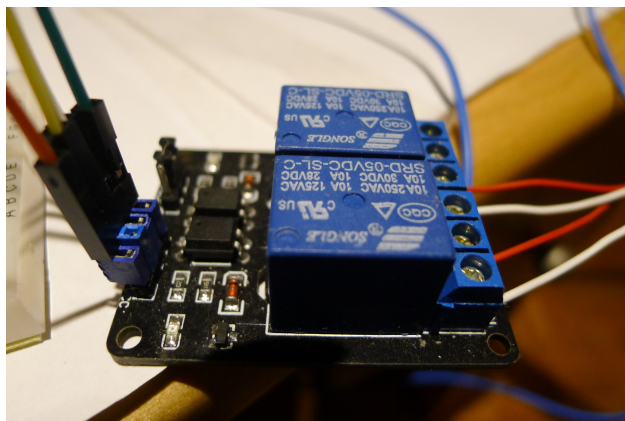
Modul (slika 2.7) napajamo z napajalnikom (9V, 0.6A) podjetja TP-Link. Vanj smo vstavili SIM-kartico ponudnika mobilnih storitev - Bob.



Slika 2.: Komunikacijski modul GSM/GPRS IcomSat SIM900 z anteno [8].

2.2.3 Modul z dvokanalnim relejem

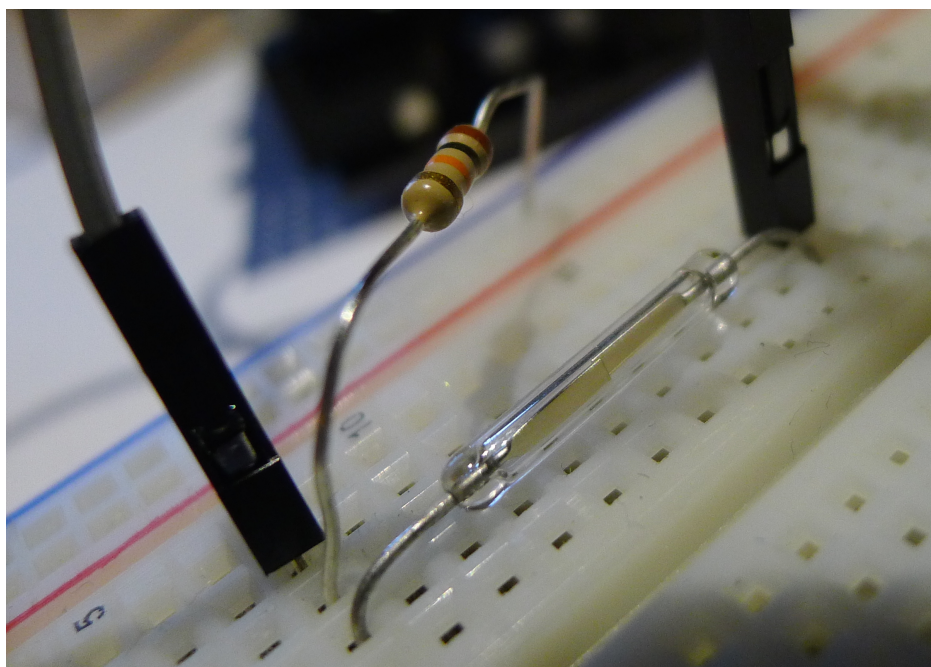
Za odpiranje in zapiranje vrat smo uporabili dvokanalni rele modul. Modul priključimo namesto ročnih stikal, ki služita za kontrolo motorja s strani uporabnika drsnih vrat. Releja je potrebno uporabiti kot mirovno razklenjen kontakt (NO). Prvi rele je namenjen popolnemu odpiranju in zapiranju vrat, drugi rele pa delnemu zapiranju in odpiranju vrat. Delno odpiranje vrat služi za prehod pešcev, koles ali motorjev. Za vhodno napetost smo uporabili napetostni izhod (5V) vgrajenega sistema, upravljamo pa ju z digitalnima izhodoma 5 (prehod za vozila) in 6 (prehod za pešce) mikrokrmilniškega sistema.



Slika 2.7: Modul z dvokanalnim relejem

2.2.4 Tipalo položaja vrat – »Reed« stikalo

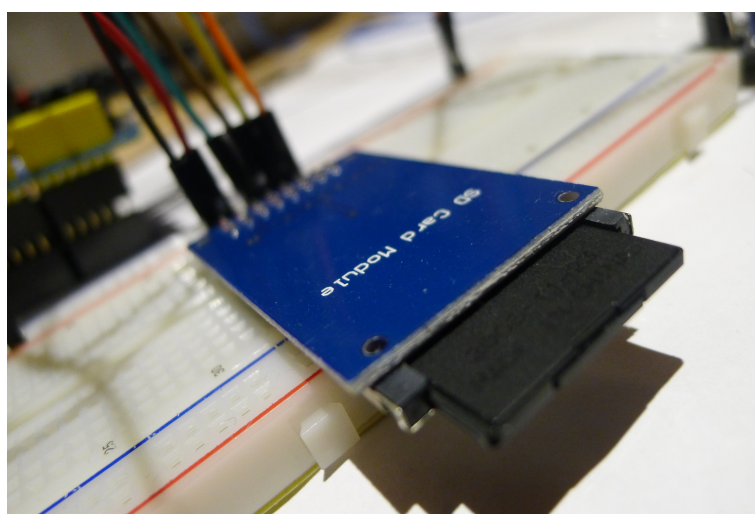
Ker s kontrolne plošče motorja BX-78 ni moč razbrati ali so vrata zaprta ali odprta, smo uporabili »reed« stikalo. Na stebričku ob motorju bo nameščeno »reed« stikalo, na koncu vrat pa magnet. Ko se bo magnet približal stikalu, se bo kontakt sklenil in skozi stikalo bo stekel električni tok. To bomo prebrali z digitalnim vhodom 3 na našem Arduinu. Pred »reed« stikalom smo zaradi omejitve električnega toka vezali še 200-ohmski upor.



Slika 2.8: Reed stikalo na testni ploščici

2.2.5 Modul z bralnikom SD-kartice

Za branje kartice za shranjevanje podatkov smo uporabili modul z bralnikom SD spominskih kartic, ki je prikazan na sliki 2.9. Priključili smo ga na testno ploščico in ter povezaliz mikrokrmilniškim sistemom Arduino. Na SD-kartici bodo shranjeni podatki o telefonskih številkah, ki lahko upravljajo z napravo. V tekstovni datoteki, ki se nahaja na SD-kartici je na prvem mestu zapisana tudi telefonska številka skrbnika, ki jo lahko tudi preprosto zamenjamo. Številke so ločene z vejicami. Bralec kartic napajamo z 3.3V ima 7 priključkov. Za shranjevanje podatkov smo uporabili 1 GB Micro SD-kartico in ustrezen »SD Card adapter«. Telefonska številka je ob vzpostavitvi sistema vedno ista in jo kasneje lahko spremenimo.



Slika 2.9: Modul z bralnikom SD-kartice

Poglavje 3 Pregled programskih tehnologij in orodij

Projekt je sestavljen iz mobilne aplikacije za mobilne telefonske aparate z operacijskim sistemom Android in kode, ki jo naložimo na mikrokrmilniški sistem Arduino, ki predstavlja osnovo vgrajenega sistema. Za izdelavo mobilne aplikacije smo uporabili okolje Android Studio, za programiranje vgrajenega sistema pa Arduino IDE. Najprej smo se lotili programiranja Arduino kode, kjer smo preizkusili vse možnosti prejemanja in pošiljanja SMS-sporočil. Ko smo zaključili s tem delom projekta, smo se osredotočili na izdelavo mobilne aplikacije.

3.1 Razvojno okolje Arduino IDE – C++

Za programiranje Arduina smo uporabili programsko orodje Arduino IDE [17], ki je programsko orodje za prenašanje izvedljivih programov na mikrokrmilniški sistem. Omogoča dodajanje C/C++ knjižnic, nalaganje skript na Arduino, omogoča pa tudi dodajanje že napisanih različnih skript za preproste projekte. Kot programski jezik uporablja C++ [16]. Arduino IDE je dostopen za operacijske sisteme Windows, OSX ter Linux [5]. Ko končamo s programiranjem, skripto naložimo na Arduino. Za branje datotek iz SD-kartice smo uporabili knjižnico SD.h [14], ki nam omogoča preprosto upravljanje z datotekami z uporabo funkcij knjižnice. Za komunikacijo in prikaz podatkov iz modula GSM smo uporabili knjižnico SoftwareSerial [12]. Za to knjižnico smo se odločili, ker smo njeno uporabo največkrat zasledili pri primerih na spletu. Na začetku smo imeli precej težav, saj smo na modulu GSM pozabili prestaviti stikali, ki določata priključka na katera je modul priključen na Arduino. Knjižnica SoftwareSerial smo uporabili tudi za prikaz komunikacije med modulom ter Arduinom v konzoli (»Serial Monitor«).

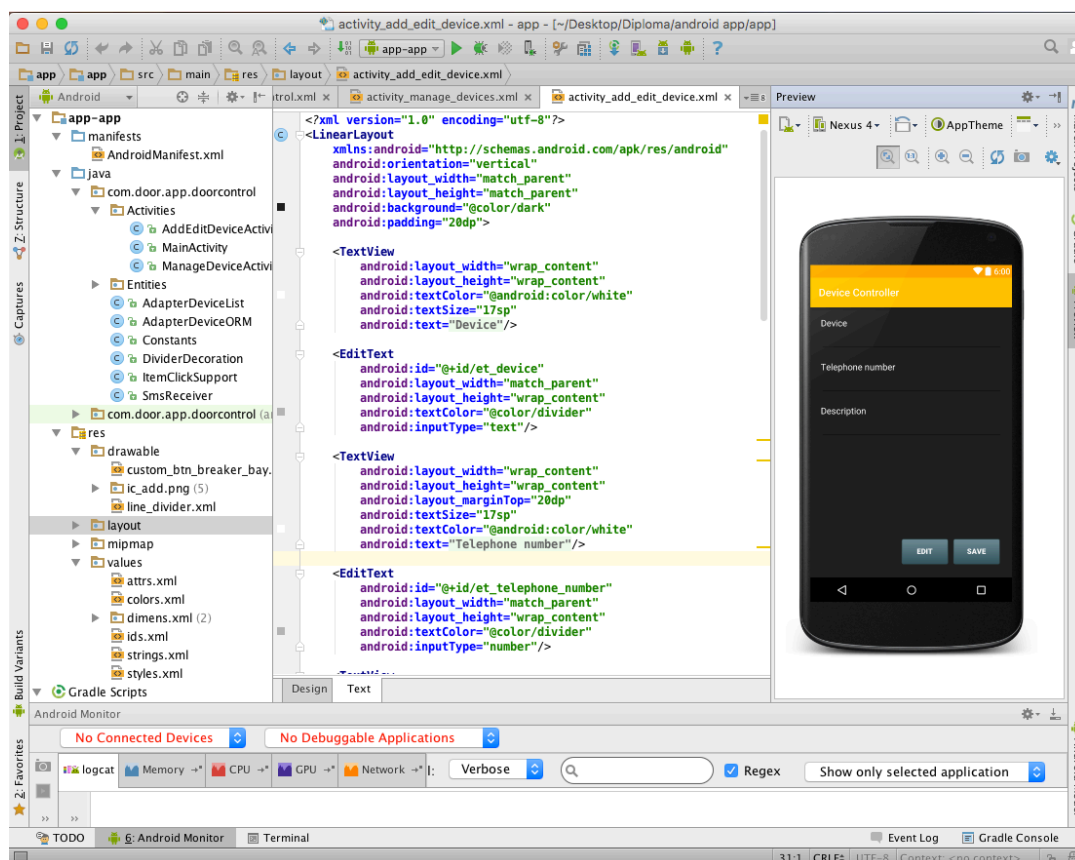


Slika 3.1: Slika ekrana z razvojnim okoljem Arduino IDE.

3.2 Razvojno okolje Android Studio – Java, XML, HTML in Sugar ORM

Za razvoj aplikacije za mobilni telefon z operacijskim sistemom Android smo uporabili razvojno okolje Android Studio [6].

Za programiranje logike smo uporabili programski jezik Java, za oblikovanje uporabniškega vmesnika pa programska jezika XML in HTML. Za urejanje uporabniškega vmesnika smo uporabili isto razvojno okolje.



Slika 3.2: Slika ekrana s prikazom oblikovanja uporabniškega vmesnika v razvojnem okolju Android Studio.

Podatkovno zbirko kreiramo in uporabljamo s knjižnico Sugar ORM [9], ki nam omogoča generiranje poizvedb ter ukazov z njenimi funkcijami. Knjižnico smo izbrali, ker nam omogoča preprosto delo s podatkovno bazo. Če bi imeli malo bolj kompleksen primer baze bi morali uporabiti kakšno bolj razširjeno knjižnico. Knjižnica ob namestitvi aplikacije na mobilni telefon kreira podatkovno bazo SQLite [15] in jo izbriše ob odstranitvi aplikacije. V tej bazi se nahajajo vse naprave z njihovimi imeni, opisom ter telefonsko številko. Končno obliko shranimo v datoteko s končnico ».apk«, ki nam omogoča namestitev na mobilne telefone z operacijskim sistemom Android.

Poglavje 4 Funkcionalnost sistema

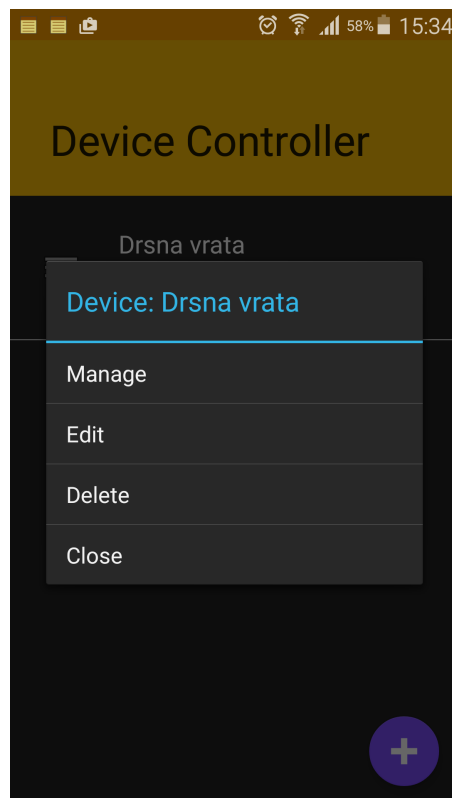
Naš sistem ima enega skrbnika sistema in več uporabnikov aplikacije, ki lahko upravljajo z drsnimi vrati. Sistem bodo v podjetju lahko uporabljali za upravljanje drsnih vrat pred skladiščem. Direktor podjetja bi lahko imel možnost dodajanja mobilnih števil, s katerih bodo lahko zaposleni upravljali z vrati. Prav tako bodo lahko iz pisarne odprli drsna vrata za prevzem ter odvoz blaga izpred skladišča.

4.1 Uporabnik aplikacije

Uporabniku aplikacije se ob zagonu prikaže seznam vseh naprav, ki jih je v preteklosti dodal. Uporabnik lahko naprave dodaja, briše ali ureja. Ob dodajanju naprave uporabnik vpiše ime naprave, njeno telefonsko številko in krajši opis.

Ob pritisku na posamezno napravo se uporabniku prikaže meni (prikazan na sliki 4.1), kjer lahko izbere, kaj želi narediti z izbrano napravo:

- Upravljaj (»Manage«):
 - Odpre novo okno, kjer lahko s to napravo upravljamo;
- Uredi (»Edit«):
 - Odpre novo okno, kjer lahko urejamo podatke o napravi (ime, telefonska številka, opis naprave);
- Izbriši (»Delete«):
 - Izbrano napravo izbriše iz aplikacije in nas preusmeri na osvežen seznam naprav;
- Zapri (»Close«):
 - Zapre meni z možnostmi in prikaže seznam naprave.



Slika 4.1: Prikaz menija za izbiro akcije.

Če uporabnik pritisne na zavihek Upravljalj se mu odpre novo okno (okno je prikazano na sliki 7.2), z več možnostmi upravljanja z napravo:

- Odpri ali zapri vrata za vozila:
 - V primeru, da je ukaz uspešno izveden se vrata odprejo ali zaprejo in ne dobimo povratne informacije. Če želimo zapreti zaprta ali odpreti odprta vrata nam sistem vrne informacijo, da so vrata že zaprta oziroma odprta.
- Odpri ali zapri vrata za pešce oziroma delno odpiranje vrat:
 - Kadar je ukaz uspešno izveden, se vrata odprejo ali zaprejo in ne dobimo povratne informacije. V primeru, da so vrata zaprta in jih želimo zapreti, dobimo povratno informacijo, da tega ni mogoče izvesti. Če pa so vrata delno odprta situacije sistem ne more preveriti, zato o tem ne dobimo podatka.
- Preveri stanje vrat:
 - Sistem nam pošlje povratno informacijo o stanju vrat, in sicer, če so vrata zaprta ali odprta.
- Preveri stanje fotocelic:
 - Sistem nam pošlje povratno informacijo, ali so celice prekrите ali ne.
- Preveri povezavo z napravo:
 - Sistem nam pošlje povratno informacijo, če se je aplikacija uspešno povezala s sistemom. V kolikor telefonske številke uporabnika še ni v sistemu, uporabnik odgovora sistema ne bo prejel.

4.2 Skrbnik sistema

Skrbnik sistema je ena oseba, ki lahko upravlja s telefonskimi številkami uporabnikov, ki imajo dovoljenje za upravljanje s sistemom za odpiranje vrat. S tem smo naredili sistem dokaj varen, saj omogoča izvajanje ukazov samo telefonskim številkam, ki so shranjene v sistemu. Do vdora v sistem bi lahko prišlo samo tako, da nekdo uporabniku aplikacije ukrade telefon in upravlja z vrati. V prihodnosti nameravamo aplikacijo zaščititi tudi pred tem načinom zlorabe. Številka skrbnika sistema je ob vzpostavitvi sistema zapisana v tekstovni datoteki. Skrbnik lahko zahteva seznam vseh števil, jih briše, dodaja ali spremeni številko skrbnika sistema. Ukaze pošilja s pomočjo SMS-sporočil na telefonsko številko, ki jo ima naprava.

Skrbnik sistema lahko pošilja naslednje ukaze:

- “getAll”:
 - S tem ukazom pridobimo vse telefonske številke, ki so trenutno shranjene v sistemu. Naprava nazaj pošlje telefonske številke ločene z vejico.
- “add +38640404040”:
 - Ukaz se uporablja za dodajanje telefonske številke v sistem, ki jo podamo kot prvi argument
- “rem +3864044040”:
 - Ukaz se uporablja za brisanje telefonske številke, ki jo podamo kot prvi argument.
- “chMan +386404040”:
 - S tem ukazom spremenimo številko skrbnika sistema. Njegovo številko podamo kot prvi argument.

Poglavje 5 Izdelava prototipa sistema

Za izdelavo prototipa sistema smo uporabili več komponent:

- Mikrokontrolniški sistem Arduino UNO,
- Motor Came BX-78,
- Bralnik SD-kartice,
- Dvo-kanalni rele,
- Modul GSM za Arduino UNO,
- Reed stikalo,
- 220 ohmski upor,
- Testna ploščica,
- Kabli za povezavo komponent,
- 9 V, 0.6 A napajalnik za modul GSM,
- PC.

Najprej smo na modul GSM prispajkali daljše nožice, ki omogočijo, da lahko modul GSM povežemo na Arduino. Na modul GSM smo nastavili mostiček za branje iz priključkov 0 ter 1 na priključka 7 in 8. Potrebno je povezati tudi ozemljitvene priključke Arduina ter GSM modula. Za vklop in izklop modula pa smo povezali priključek 9 na Arduinu ter PWRKEY na modulu.

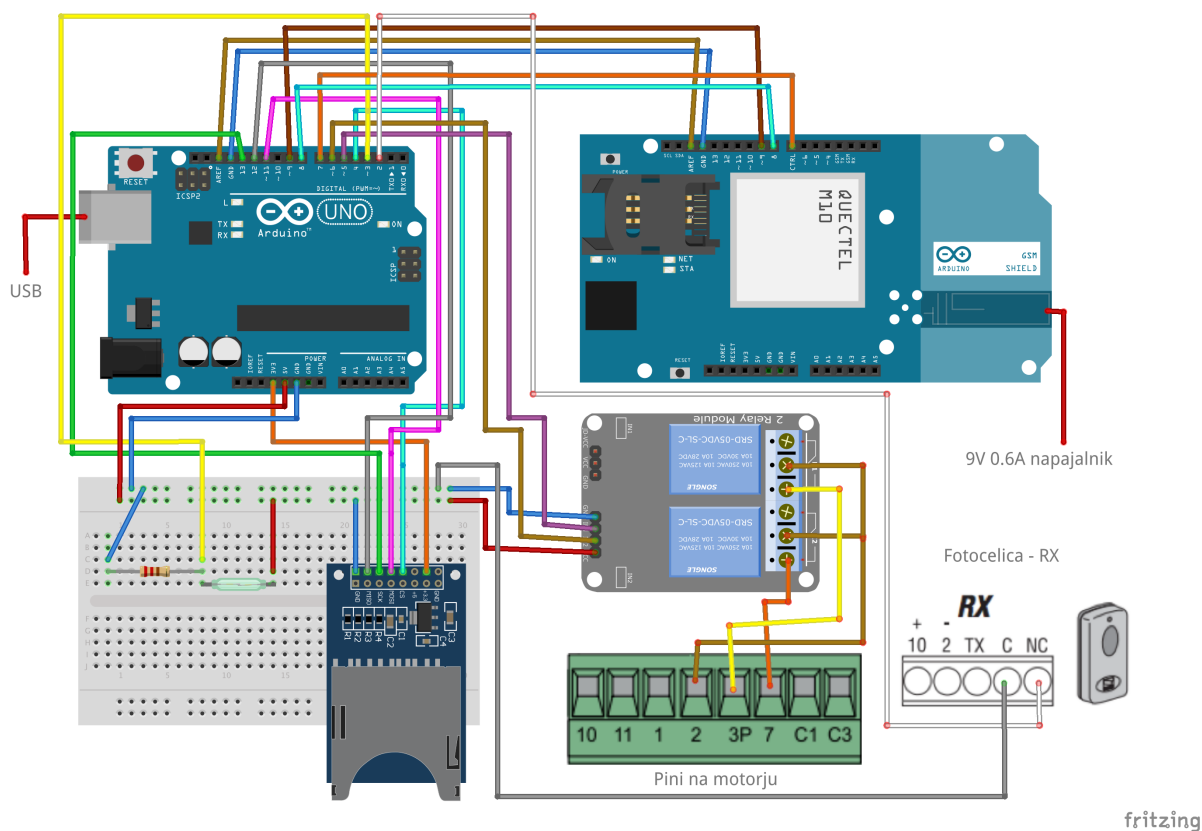
Na testno ploščico smo priključili 5V izhod in ozemljitveni priključek iz Arduina. Na testno ploščico smo dodali »Reed« stikalo ter predenj zvezali 220 ohmski upor. Za upor smo kabel peljali na digitalni priključek 3, ki bo bral stanje reed stikala. Na ploščico smo namestili tudi bralnik SD-kartice in ga ustrezno priključili na Arduino. Skica prototipa sistema je prikazana na sliki 5.2, fotografija celotnega prototipa pa na sliki 5.3.

Arduino priključek	Vhod na bralcu SD kartic
/	5V
3.3V	3.3V
GND	GND
11	MISO
13	CLK
4	CS

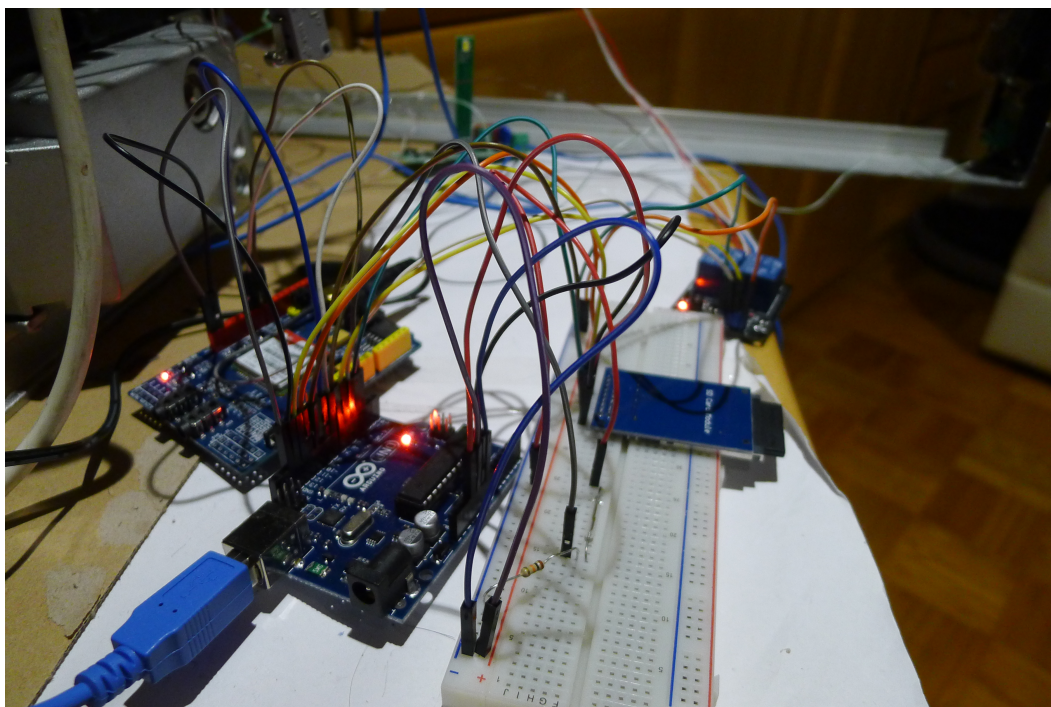
Tabela 5.1: Priklop bralca SD-kartic na Arduino

Za bralnik SD-kartic smo uporabili 3,3V izhod na Arduinu, saj je tako verjetnost motenj delovanja modulov manjša. Na testno ploščico smo priključili tudi dvo-kanalni rele. Nanj smo peljali 5V ter ozemljitveno povezavo, priključili pa smo ga na digitalna priključka 5 in 6. Iz releja smo kabla peljali na motor in ju priključili na priključka 2 ter 3P (delno odpiranje in zapiranje vrat) in 7 (odpiranje in zapiranje vrat) na kontrolni plošči motorja. Releja smo vezali kot mirovno odprti stikali, tako da se kontakt v releju sklene, ko Arduino pošlje signal prek digitalnega priključka. Za napajanje Arduino smo uporabili kabel USB, za napajanje modula GSM pa 9V napajalnik (0.6A).

Ker je izhodna napetost iz releja v fotocelicah 5V ali manj smo lahko priključek NC priklopili direktno na digitalni priključek 2. Iz fotocelic smo priključek C povezali z ozemljitvenim priključkom Arduina.



Slika 5.2: Skica prototipa sistema.



Slika 5.3: Prototip sistema.

Poglavje 6 Programska oprema vgrajenega sistema

Po priključitvi motorja vrat smo se najprej osredotočili na upravljanje motorja z Arduino ploščico. Na Arduino ploščico smo zvezali dva releja, ki bosta služila za odpiranje ter zapiranje drsnih vrat. Ko smo dosegli nemoteno delovanje vrat, smo na Arduino prikllopili modul GSM, v katerega smo vstavili SIM-kartico brez gesla. Arduino smo sprogramirali tako, da se ob prejetem SMS-sporočilu izvede zelen ukaz. V naslednji fazi smo na Arduino priključili še bralnik SD-kartic, ki služi za shranjevanje številke. Na kartici smo ustvarili tekstovno datoteko, na kateri so vpisane številke, s katerih naprava lahko prejema SMS-sporočila z ukazi. Na začetku mora biti v datoteki zapisana številka skrbnika sistema, ki lahko začne z dodajanjem številke uporabnikov drsnih vrat. Ker je tekstovna datoteka zapisana na SD-kartici se tudi ob izpadu električne energije številke ohranijo. Če skrbnik doda telefonsko številko se na konec datoteke zapiše nova številka, razen če ta številka že obstaja. Številke so med seboj ločene z vejicami. Skrbnik ne more izbrisati svoje telefonske številke, ampak jo lahko samo zamenja z drugo telefonsko številko.

Po zaključenem prvem delu naloge, smo se posvetili aplikaciji za mobilne telefone z operacijskim sistemom Android. Najprej smo sprogramirali funkcije, s katerimi lahko uporabnik dodaja, briše in ureja naprave v aplikaciji. Kasneje smo za vsako dodano napravo vključili tudi možnost njenega upravljanja. Ko uporabnik pritisne na gumb, aplikacija pošlje SMS-sporočilo in čaka na odgovor naprave. Ob prejetju SMS-sporočila iz izbrane številke, se v aplikaciji pokaže pojavno okno s povratnimi informacijami.

Tretji del naloge je bil namenjen podpori dodajanja in brisanja številke na SD-kartici. Številke iz SD-kartice lahko briše samo uporabnik, katerega številka je v datoteki napisana na prvem mestu. Ta uporabnik lahko osebo, ki je zapisana na prvem mestu z ustreznim ukazom tudi zamenja.

6.1 Upravljanje vrat

Za zapiranje, odpiranje in preverjanje stanja vrat smo uporabili dva releja in reed stikalo. Najprej smo definirali vse komponente, ki jih bomo uporabili pri naši aplikaciji:

- Rele za odpiranje ter zapiranje vrat za vozila na digitalnem priključku 5.
- Rele za delno odpiranje ter zapiranje vrat na digitalnem priključku 6.
- Reed stikalo za preverjanje stanja vrat na digitalnem priključku 3.
- Vhod za preverjanje napetosti iz fotocelic na digitalnem priključku 2.

Releja sta uporabljena kot mirovno odprti stikali, zato ju je potrebno za pošiljanje signalov motorju skleniti. To povzroči, da se vrata odprejo ali zaprejo.

Preverjanje ali so vrata odprta smo opravili z branjem stanja reed stikala. Če je reed stikalo sklenjeno pomeni, da so vrata zaprta. Če je stikalo razklenjeno pa pomeni, da so vrata odprta ali delno odprta.

```
if (msg.equals("close")) { // zapre vrata
    if (digitalRead(reedPin) == HIGH) { // preverimo če so vrata že zaprta
        SendMessage(mob, "gateAlreadyClosed"); // pošljemo SMS pošiljatelju o stanju vrat
    }
    else { // če so vrata odprta jih zapremo z relejem
        digitalWrite(RELAY_START, 0);
        delay(500);
        digitalWrite(RELAY_START, 1);
    }
}
```

Izsek kode 6.1: Zapiranje vrat.

6.2 Komunikacija z mobilnim omrežjem

Za komunikacijo z mobilnim omrežjem smo uporabili modul GSM. Modul GSM za komunikacijo uporablja priključka 7 in 8 ter uporablja serijski protokol. Ker podatkov o pisanju in branju iz priključkov 0 in 1 ne prikaže v konzoli, smo uporabili knjižnico SoftwareSerial [12], ki omogoča, da lahko z definirano spremenljivko beremo iz želenih priključkov (SoftwareSerial mySerial 7, 8). Njegovo delovanje omogočimo s pošiljanjem AT ukazov. V našem primeru bomo uporabili samo način za pošiljanje in prejemanje SMS-sporočil.

```
// indikacija moči signala
mySerial.println("AT+CSQ");
delay(2000);
ShowSerialData();

// nastavi SMS sistem v tekstovni način, ki je nasproten PDU načinu
mySerial.println("AT+CMGF=1");
delay(2000);
ShowSerialData();

// potrdi, da se je naprava registrirala v omrežje
mySerial.println("AT+CREG?");
delay(2000);
ShowSerialData();

// ukaz prepošlje sporočila na TE in nastavimo obliko sporočila tako,
// da bomo lahko kasneje iz njega pobrali sporočilo ter številko pošiljatelja
mySerial.println("AT+CNMI=1,2,0,0,0"); |
delay(2000);
ShowSerialData();
```

Izsek kode 6.2: Primer uporabe AT ukazov.

V večni zanki `loop()` [13] preverjamo, če pride kakšen podatek na priključka 7 ter 8 in nato kličemo funkcijo `ReceiveSMS()`, ki preveri, če so podatki SMS-sporočilo in jih obdela.

V funkciji `ReceiveSMS()` najprej razčlenimo SMS-sporočilo ter iz njega pridobimo vsebino sporočila in pošiljateljevo številko. Če je SMS-sporočilo prekratko ga ignoriramo. Del te funkcije je prikazan na sliki 6.3.

```
Serial.print(F("Message Text: "));
for (int x = 45; x < MsgLength; x++)
{
    MsgTxt[x - 45] = RcvdMsg[x];
    msg.concat(MsgTxt[x - 45]);
    Serial.print(MsgTxt[x - 45]);
}
Serial.println(F(""));
String secondNum = "";

if (msg.startsWith("add")) { // dodajanje številke za admina
    secondNum = msg.substring(4, msg.length());
    addNewNumber(mob, secondNum);
}

if (msg.startsWith("rem")) { // brisanje številke za admina
    secondNum = msg.substring(4, msg.length());
    removeNumber(mob, secondNum);
}

if (msg.startsWith("getAll")) { // pridobi vse številke za admina
    getAllNumbers(mob);
}

if (msg.startsWith("chm")) { // zamenja številko admina / upravljalca
    secondNum = msg.substring(4, msg.length());
    changeManager(mob, secondNum);
}

if (msg.length() > 2) { // se znebimo kratkih sporočil
    if (checkIfNumExists(mob)) { // preverjanje če obstaja številka, če obstaja izvaja naprej
        if (msg.equals("open")) { // odpre vrata
            if (digitalRead(reedPin) == LOW) {
                SendMessage(mob, "gateAlreadyOpen");
            }
        }
    }
}
```

Izsek kode 6.3: Pridobivanje teksta iz prejetega SMS-sporočila ter klici funkcij za nadaljno obdelavo.

6.3 Branje in izvajanje ukazov uporabnika

Uporabnik sistema lahko naenkrat pošlje samo en ukaz. Če v sistem naenkrat pride več ukazov, sistem obdela samo prvega, ostale pa ignorira. To smo storili zaradi nevarnosti ponovnega zapiranja ali odpiranja vrat ob vstopu vozil ali pešcev. V prihodnosti imamo namen implementirati tudi možnost opozarjanja uporabnikov, da je bil njihov ukaz zavrnjen, ker je imel sistem v obdelavi ukaz drugega uporabnika. Posredovali bi mu lahko tudi številko uporabnika, katerega ukaz je bil poslan prej. Sistem uporabniku na nekatere ukaze oziroma poizvedbe vrne povratno informacijo. Na Arduino smo priključili tudi bralnik SD-kartice, ki nam služi za shranjevanje tistih telefonskih števil, ki imajo dovoljenje za upravljanje z napravo. Številke so zapisane v tekstovni datoteki in ločene z vejicami. Za branje in pisanje na SD kartico smo uporabili knjižnico SD.h [14].

```
boolean checkIfNumExists(String num) {
    boolean exists = false;
    myFile = SD.open("stevilke.txt"); // odpremo datoteko
    int i = 1;
    if (myFile) {
        Serial.println(F("stevilke.txt:"));
        while (myFile.available()) {
            String line = myFile.readStringUntil(','); // beremo do vejice
            Serial.println(line);
            if (line == num) { // preverimo če je vrstica enaka številki v argumentu
                exists = true; // če je enaka nastavimo vrednost na true
                Serial.println(F("Številka obstaja"));
            } else {
                exists = false; // če ni enaka nastavimo vrednost na false
            }
        }
    } else {
        Serial.println(F("napaka pri branju datoteke stevilke.txt"));
    }
    myFile.close();
    return exists; // vrnemo vrednost
}
```

Izsek kode 6.4: Preverjanje, če številka na SD kartici obstaja.

Ob vsakem prejemu SMS-sporočila, najprej z funkcijo »checkIfNumExists« preverimo, ali pošiljateljeva telefonska številka obstaja v našem sistemu. Funkcija je prikazana v izseku kode 6.4. V primeru, da nam funkcija vrne pozitiven odgovor, se lotimo nadaljnje obdelave sporočila.

Sistem lahko obdela več ukazov s strani uporabnika po preverjanju veljavnosti telefonske številke:

- Prejeto sporočilo: "open" – odpiranje vrat:
 - Sistem najprej preveri, če so vrata že odprta in v tem primeru uporabniku pošlje SMS-sporočilo, da odprtih vrat ni mogoče odpreti. Če so vrata zaprta,

sistem vrata odpre. Uporabnik v tem primeru ne dobi povratne informacije. Če so vrata že odprta, sistem uporabniku pošlje SMS-sporočilo "gateAlreadyOpen".

- Prejeto sporočilo: "close" – zapiranje vrat:
 - Sistem najprej preveri, če so vrata že zaprta in v tem primeru odgovori uporabniku, da so vrata že zaprta. Če so vrata odprta, sistem zapre vrata. Uporabnik v tem primeru ne dobi povratne informacije. Če so vrata že zaprta, sistem uporabniku pošlje SMS-sporočilo "gateAlreadyClosed".
- Prejeto sporočilo: "openPed" – delno odpiranje vrat za pešce:
 - Sistem izvede isto operacijo, kot če prejme ukaz "open". Razlikuje se samo v tem, da za odpiranje vrat uporabi rele, ki je namenjen delnemu odpiranju vrat.
- Prejeto sporočilo: "clsPed" – zapiranje vrat za pešce:
 - Sistem izvede isto operacijo, kot če prejme ukaz "close". Preveri tudi če so vrata že zaprta, v nasprotnem primeru pa obvesti uporabnika. Za odpiranje vrat uporabi rele, ki je namenjen delnemu zapiranju vrat. Če so vrata že zaprta, sistem uporabniku pošlje SMS-sporočilo "gateAlreadyClosed".
- Prejeto sporočilo: "gateSt" – preveri stanje vrat:
 - Sistem preveri če so vrata zaprta ali odprta. Preveri stanje »reed« stikala in če je stikalo sklenjeno uporabniku pošlje povratno informacijo, da so vrata zaprta. V drugem primeru uporabniku pošlje informacijo, da so vrata odprta ali delno odprta. Če so vrata zaprta, sistem uporabniku pošlje sporočilo "gateClosed", v obratnem primeru pa "gateOpen".
- Prejeto sporočilo: "cellSt" – preveri stanje fotocelic:
 - Sistem na fotocelicah motorja preveri ali so fotocelice prekrите. Po analizi sistem uporabnika obvesti o stanju fotocelic. Če so fotocelice prekrите, sistem uporabniku pošlje SMS-sporočilo "photocellCovered", v obratnem primeru pa "photocellOk".
- Prejeto sporočilo: "connSt" – preveri, če se je možno povezati:
 - Sistem preveri, če je uporabnikova številka v sistemu in uporabniku odgovori s sporočilom "connectionOk".

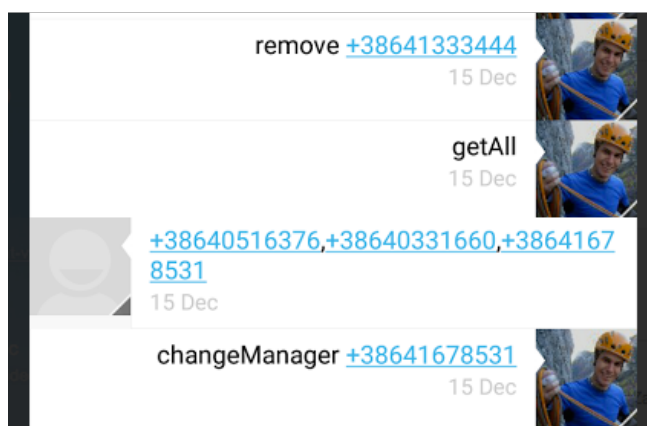
6.4 Branje in izvajanje ukazov skrbnika

Uporabnik ima več možnosti upravljanja s telefonskimi številkami, ki so shranjene na SD-kartici. Ob klicu vsake funkcije sistem najprej preveri, če se v tekstovni datoteki, kjer so zapisane telefonske številke, na prvem mestu nahaja telefonska številka pošiljatelja in šele nato izvede ukaz. Sistem lahko sprejme štiri različne ukaze skrbnika:

- Prejeto sporočilo: "add telefonska_številka":
 - Doda telefonsko številko na seznam, ki jo podamo kot argument. S funkcijo »addNewNumber(mob, secondNum)« najprej preverimo, če se na prvem

mestu nahaja telefonska pošiljatelja in nato dodamo novo telefonsko številko na seznam števil, če ta še ne obstaja.

- Prejeto sporočilo: "rem telefonska_števila":
 - Odstrani telefonsko številko iz seznama, ki jo podamo kot argument. S funkcijo »removeNumber(mob, secondNum)« najprej preverimo, če se na prvem mestu nahaja telefonska pošiljatelja, in nato odstranimo telefonsko številko, ki jo podamo kot drugi argument.
- Prejeto sporočilo: "getAll":
 - Vrne seznam vseh števil, ločenih z vejicami. S funkcijo »getAllNumbers(mob)« najprej preverimo, če se na prvem mestu nahaja telefonska pošiljatelja in nato pošiljatelju posredujemo seznam števil iz tekstovne datoteke. Oblika poslanega sporočila je identična vsebini tekstovne datoteke.
- Prejeto sporočilo: "chMan telefonska_števila":
 - Spremeni telefonsko številko tistega, ki lahko spreminja seznam števil. S funkcijo »changeManager(mob, secondNum)« najprej preverimo, če se na prvem mestu nahaja telefonska pošiljatelja, ki smo jo podali kot prvi argument. Po preverjanju na seznamu telefonskih števil v tekstovni datoteki na prvem mestu zamenjamo telefonsko številko s številko, ki smo jo podali kot argument.



Slika 6.5: Primer uporabe – zaporedje skrbnikovih SMS sporočil.

Poglavje 7 Mobilna aplikacija za upravljanje drsnih vrat

Ker je upravljanje naprave s pomočjo SMS-sporočil precej nepregledno in zamudno, smo se odločili, da bomo izdelali aplikacijo, ki bo sposobna pošiljati in prejemati SMS-sporočila. Izbrali smo operacijski sistem Android, ki je najbolj razširjen med pametnimi telefoni.

7.1 Uporabniški vmesnik mobilne aplikacije

Najprej smo izdelali uporabniški vmesnik. Ko zaženemo aplikacijo se nam prikaže seznam naprav, kjer je prikazano ime naprave ter njen opis. Za prikaz naprav smo uporabili knjižnico RecyclerView [11].

Implementirali smo možnost klika na napravo, ki nam odpre prikazno okno s štirimi možnostmi. Možne izbire so brisanje, urejanje ter upravljanje naprave.

Ob kliku na gumb za brisanje naprave nas aplikacija najprej vpraša, če napravo res želimo izbrisati in nas ob potrditvi preusmeri nazaj na okno kjer so prikazane vse naprave.

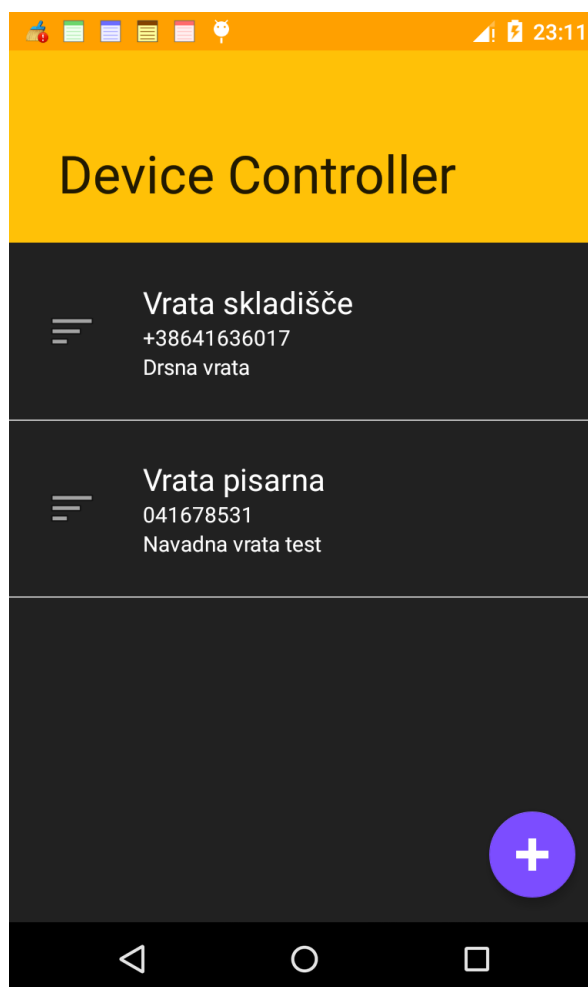
Ob kliku na upravljanje naprave se nam odpre novo okno s sedmimi gumbi. Gumbi nam omogočajo odpiranje ter zapiranje vrat za vozila in pešce, preverjanje statusa vrat, preverjanje statusa fotocelic ter preverjanje povezave med mobilnim telefonom ter vgrajenim sistemom.

V desni spodnji kot smo postavili plavajočo ikono, ki nam ob kliku omogoča dodajanje naprav. Ob kliku na ikono se nam odpre novo okno s tremi tekstovnimi polji za vpis imena ter opisa naprave in njene telefonske številke. Na tem smo dodali še dva gumba, ki nam omogočata shranjevanje ter preklic dodajanja nove naprave. Ob kliku na gumba nas v obeh primerih aplikacija preusmeri na prvo okno. Ob kliku na gumb za shranjevanje se osveži tudi prikaz RecyclerView-a z novo dodano napravo. Če kliknemo na gumb za urejanje se nam prikaže podoben vmesnik kot za dodajanje naprave, z razliko, da nam aplikacija napolni tekstovna polja s podatki izbrane naprave.

7.2 Dodajanje, urejanje in brisanje naprav

Odločili smo se, da za prikaz naprav uporabimo gradnik recyclerView (prikazan na sliki 7.1), saj nam bo to omogočilo dodajanja več različnih naprav. Naprave bomo shranjevali v podatkovni bazi, ki smo jo ustvarili s pomočjo knjižnice Sugar ORM [9]. Uporabnik lahko novi napravi določi ime naprave, njeno telefonsko številko in doda krajši opis naprave. V

primeru dodajanja nove naprave se ob pritisku na gumb za shranjevanje vnos zapiše v bazo, potem pa nas aplikacija preusmeri na osvežen seznam vseh naprav.



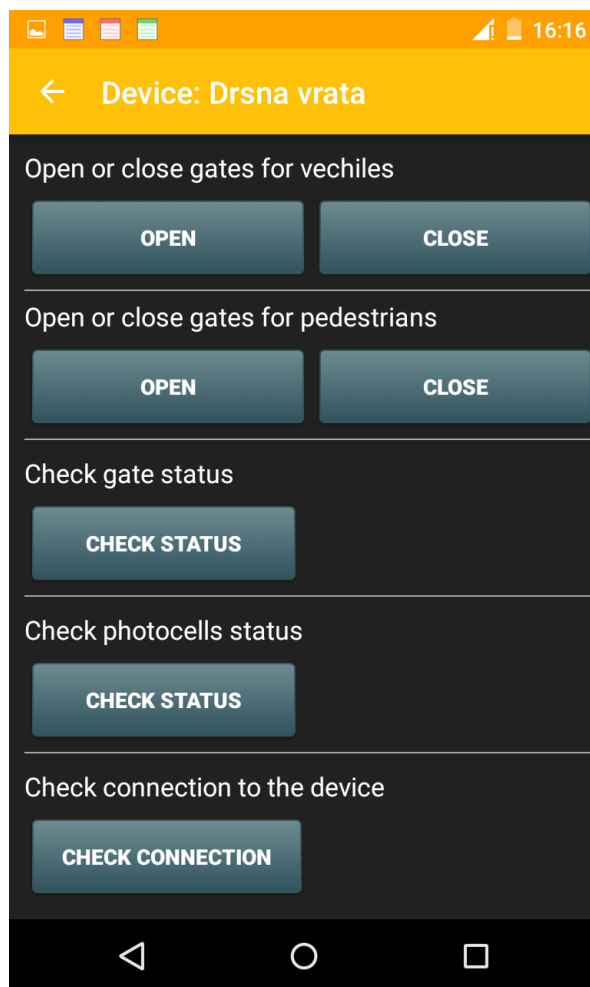
Slika 7.1: Pregled naprav v aplikaciji.

V kolikor želimo napravo urediti ali izbrisati se nam s pritiskom na izbrano napravo pokaže spustni meni, kjer imamo na izbiro več možnosti. Ob izbrisu naprave se izbran vnos izbriše iz podatkovne baze, aplikacija pa nas preusmeri nazaj na osvežen seznam vseh naprav. Lastnosti naprave lahko tudi urejamo, in sicer lahko spremenimo njeno ime, telefonsko številko ali kratek opis. Izbran vnos se ob pritisku na gumb posodobi v podatkovni bazi, aplikacija pa nas preusmeri na osvežen seznam vseh naprav.

7.3 Upravljanje naprav

V aplikacijo smo implementirali tudi možnost upravljanja z dodanimi napravami. Za prejemanje sporočil smo uporabili knjižnico `BroadcastReceiver` [10], ki se zažene takrat, ko imamo odprto okno za upravljanje izbrane naprave. Knjižnica služi za poslušanje, ko prejmemo novo sporočilo. V aplikaciji se nam ob prejemu sporočila izbrane naprave prikaže

obvestilo v obliki pojavnega okna. Če je sporočilo prišlo iz druge telefonske številke aplikacija sporočilo ignorira. Temu delu aplikacije smo morali omogočiti tudi možnost prejemanja in pošiljanja SMS-sporočil. Do tega dela aplikacije pridemo tako, da pritisnemo na napravo in izberemo zavihek “Manage”. Odpre se nam novo okno, ki nam ponuja več možnosti za upravljanje z vrati (okno prikazano na sliki 7.2).



Slika 7.2: Upravljanje z napravo v aplikaciji.

Ime naprave je napisano v statusni vrstici. Ob kliku na gumb aplikacija pošlje SMS-sporočilo na napravo, ki smo jo izbrali (prikazano na izseku kode 7.1 ter 7.2). Če je bilo sporočilo uspešno poslano, nam naprava prikaže pojavno okno (prikazano na sliki 7.3), kjer je napisano ime naprave, čas in sporočilo, ki je bilo poslano. Ko dobimo povratno informacijo od izbrane naprave, se je vidno pojavno okno s prejetimi informacijami. Če SMS-sporočilo prejmemo iz druge telefonske številke, aplikacija sporočilo enostavno ignorira.

```

@Override
public void onClick(View view) {
    Context context = getApplicationContext();
    SimpleDateFormat sdf = new SimpleDateFormat("d.M.yyyy - H:mm:ss");
    String currentDateandTime = sdf.format(new Date());

    if (view.getId() == btnOpenVechile.getId()) { // odpri vrata za vozila
        sendSMSMessage("open");
        displayAlertSent("Gates are opening. You can check their status with 'Checking gate status'.");
    }
    if (view.getId() == btnCloseVechile.getId()) { // zapri vrata za vozila
        sendSMSMessage("close");
        displayAlertSent("Gates are closing. You can check their status with 'Checking gate status'.");
    }
    else if (view.getId() == btnOpenPedestrian.getId()) { // odpri vrata za pešce
        sendSMSMessage("opnPed");
        displayAlertSent("Gates are opening for pedestrian access. You can close them with close button for pedestrians.");
    }
    else if (view.getId() == btnClosePedestrian.getId()) { // zapri vrata za pešce
        sendSMSMessage("clsPed");
        displayAlertSent("Gates are closing. You can check their status with 'Checking gate status'.");
    }
    else if (view.getId() == btnGateStatus.getId()) { // preveri stanje vrat
        sendSMSMessage("gateSt");
        displayAlertSent("We are checking gate status. You should be notified soon.");
    }
    else if (view.getId() == btnPhotocellStatus.getId()) { // preveri stanje fotocelic
        sendSMSMessage("cellSt");
        displayAlertSent("We are checking photocells status. You should be notified soon.");
    }
    else if (view.getId() == btnConnectionStatus.getId()) { // preveri stanje povezave
        sendSMSMessage("connSt");
        displayAlertSent("We are checking connection status. You should be notified soon, if not, connection is not working.");
    }
}
}

```

Izsek kode 7.1: Del kode, ki skrbi za pošiljanje ukazov v obliki SMS-sporočila vgrajenemu sistemu.

```

protected void sendSMSMessage(String message) {

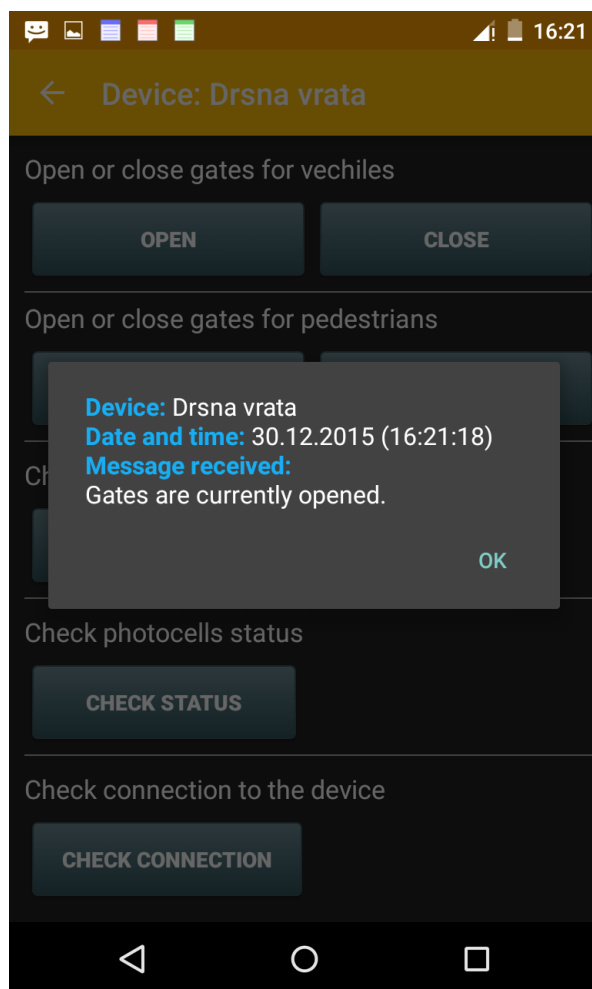
    String phoneNo = mDevice.getTelephoneNumber();

    try {
        SmsManager smsManager = SmsManager.getDefault();
        smsManager.sendTextMessage(phoneNo, null, message, null, null);
    }

    catch (Exception e) {
        Toast.makeText(getApplicationContext(), "SMS failed, please try again.", Toast.LENGTH_LONG).show();
        e.printStackTrace();
        Log.getStackTraceString(e);
    }
}

```

Izsek kode 7.2: Funkcija, ki jo uporabljamo za pošiljanje SMS-sporočil.



Slika 7.3: Pojavno okno ob prejemu SMS-sporočila.

Poglavje 8 Sklepne ugotovitve

V sklopu diplomske naloge smo razvili sistem za upravljanje motorja za drsna vrata preko SMS-sporočil in mobilno aplikacijo, s katero uporabniku olajšamo upravljanje motorja. V mobilni aplikaciji lahko preprosto dodajamo nove naprave, ki ostanejo vidne do izbrisa aplikacije. Pošiljanje SMS-sporočil je zelo preprosto, saj ukaze pošiljamo enostavno s pomočjo gumbob ob pritisku na upravljanje naprave. Pošiljanje SMS-sporočil preko standardnega vmesnika je precej nerodno, saj si mora uporabnik zapomniti ukaze in vedeti kaj pomeni povratna informacija poslana iz vgrajenega sistema. Ob prejetju SMS-sporočila aplikacija sporočilo obdela in ga prikaže uporabniku prijazno. Implementirali smo tudi sistem, ki omogoča varno uporabo aplikacije, saj za vsako telefonsko številko preveri, če je dodana v sistem in šele nato izvede ukaz.

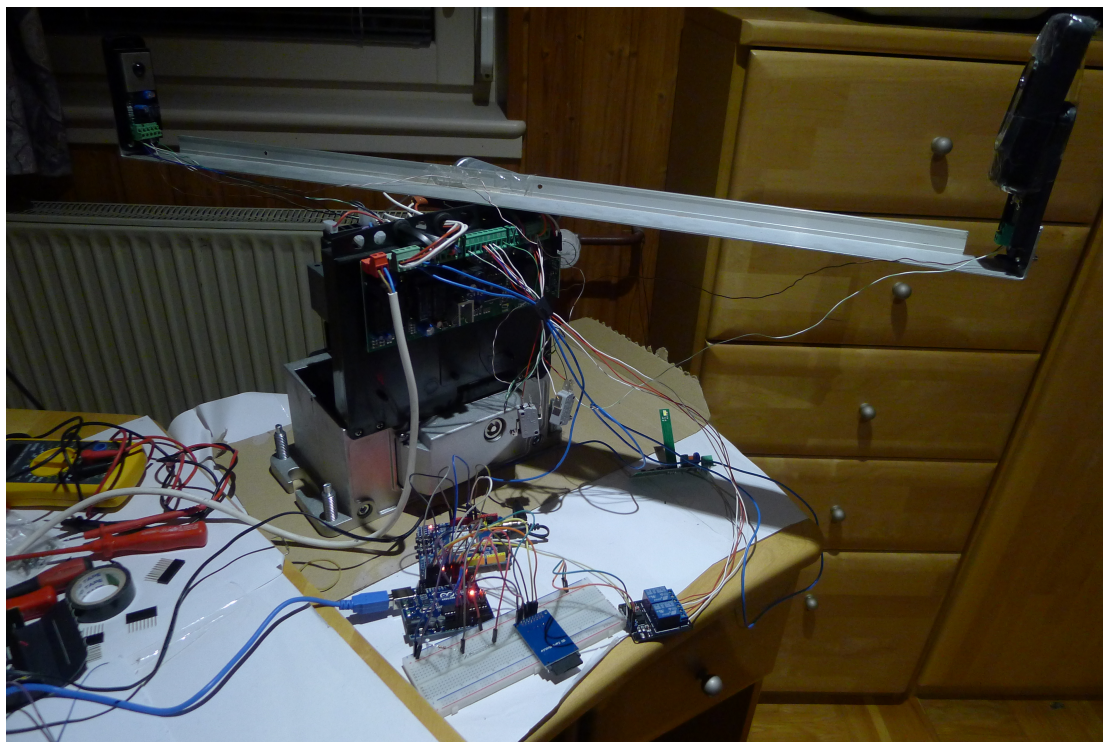
Izpolnili smo vse cilje, ki smo si jih zadali pred začetkom izdelave diplomske naloge. Največ težav nam je povzročala vezava motorja in priključitev komponent nanj. Prvič smo uporabili tudi mikrokrmilniški sistem Arduino, ki je v celoti izpolnil naša pričakovanja.

Izvedba sistema je imela tudi ekonomsko naravnan cilj. Stroškovno učinkovitost smo poizkušali doseči s ceno sistema, ki bi bil dostopen vsakomur, ki bi se odločil za nakup vrat. Cena komponent za naš prototipni sistem je znašala okoli 55 evrov, predpostavljamo pa, da bi se z nekaj optimizacije in izbiro cenejših komponent še precej znižala.

Možnih je tudi več nadgradenj sistema. V prihodnosti nameravamo aplikacijo prilagoditi, da bi lahko dodajali tudi druge naprave, kot so potopni stebrički, zapornice, dvižna vrata in poljubne druge naprave. Glede na to, da nekateri uporabniki ne želijo uporabljati SIM-kartice zaradi mesečnih stroškov, bi sistem lahko prilagodili tudi za delovanje prek brezžičnega interneta.

V podjetju so bili z prikazom delovanja prototipa zadovoljni, zato smo se odločili, da bomo prvi testni sistem namestili na drsna vrata skladišča njihovega podjetja. Če bi se zadeva izkazala za uporabno, bi lahko izdelek po preverjenem delovanju tudi tržili.

Ob izdelavi diplomske naloge smo se veliko naučili, predvsem pa dobili motivacijo za dopolnitev tega projekta in izdelavo še kakšnega podobnega sistema.



Slika 8.1: Končni prototipni sistem.

Poglavje 9 Literatura

- [1] Motorni pogon BX-78. Palisada d.o.o.. [Online]. Dosegljivo: <http://www.palisada.si/wp-content/uploads/2012/01/BX.pdf>.
- [2] Came Cancelli Automatici S.p.a., "Installation manual BX-74 / BX-78", str.1-24, 2014.
- [3] Arduino Uno. Genuino. [Online]. Dosegljivo: <https://www.arduino.cc/en/Main/ArduinoBoardUno>.
- [4] IComSat. Itead studio. [Online]. Dosegljivo: <http://wiki.iteadstudio.com/IcomSat>.
- [5] What is Arduino?. Arduino. [Online]. Dosegljivo: <https://www.arduino.cc/en/Guide/Introduction>.
- [6] Android Studio Overview. Android. [Online]. Dosegljivo: <http://developer.android.com/tools/studio/index.html>.
- [7] Arduino Uno r. 3. Wikipedia. [Online]. Dosegljivo: https://upload.wikimedia.org/wikipedia/commons/3/38/Arduino_Uno_-_R3.jpg.
- [8] Icomsat v1.1 SIM900 GSM/GPRS shield for Arduino. Itead. [Online]. Dosegljivo: https://www.itead.cc/media/catalog/product/i/m/im120417009_1_2.jpg.
- [9] Knjižnica Sugar ORM, Satyan. [Online]. Dosegljivo: <http://satyan.github.io/sugar/>.
- [10] BroadcastReceiver,.Arduino. [Online]. Dosegljivo: <http://developer.android.com/reference/android/content/BroadcastReceiver.html>.
- [11] RecyclerView. Arduino [Online]. Dosegljivo: <http://developer.android.com/reference/android/support/v7/widget/RecyclerView.html>.
- [12] SoftwareSerial, Arduino. [Online]. Dosegljivo: <https://www.arduino.cc/en/Reference/SoftwareSerial>.

- [13] Loop function, Arduino. [Online]. Dosegljivo: <https://www.arduino.cc/en/Reference/Loop>.
- [14] Arduino knjižnica SD, Arduino. [Online]. Dosegljivo: <https://www.arduino.cc/en/Reference/SD>.
- [15] SQLite, Wikipedia. [Online]. Dosegljivo: <https://en.wikipedia.org/wiki/SQLite>.
- [16] C++, Wikipedia. [Online]. Dosegljivo: <https://sl.wikipedia.org/wiki/C++>.
- [17] Arduino Software (IDE). Arduino. [Online]. Dosegljivo: <https://www.arduino.cc/en/Guide/Environment>.

